Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (currently amended) ~~In a~~ A microprocessor ~~computer system including a processor~~ having a plurality of registers, ~~a method for generating an aligned vector of first width from two second width vectors for single instruction multiple data (SIMD) processing,~~ comprising ~~the steps of~~:

a first processing unit that

in response to a first load instruction loads ~~loading~~ a first ~~vector~~ plurality of data bytes ~~from a memory unit~~ into a first register, ~~wherein the first vector contains a first byte of the aligned vector to be generated;~~

in response to a second load instruction loads ~~loading~~ a second ~~vector~~ plurality of data bytes ~~from the memory unit~~ into a second register, and [[;]]

in response to an alignment instruction determines ~~determining~~ a starting data byte in the first register, wherein the starting data byte specifies ~~the~~ a first data byte of ~~the~~ an aligned vector, [[;]] extracts ~~extracting~~ the aligned vector from the first register and the second register beginning from ~~the~~ a first bit in the starting byte of the first register continuing through bits in the second register, [[;]] and replicates ~~replicating~~ the aligned vector into a third register such that the third register contains a plurality of data elements aligned for ~~SIMD~~ single instruction multiple data (SIMD) processing; and

a SIMD processing unit coupled to the first processing unit that operates on the aligned vector.

2-48. (canceled)

49. (currently amended) A microprocessor having a plurality of registers, ~~method for generating an aligned vector from two source vectors for single instruction multiple data (SIMD) processing,~~ comprising ~~the steps of~~:

a first processing unit that

~~(1) loading~~ in response to a first load instruction loads a first ~~source vector~~ plurality of data bytes into a first register, [[;]]

~~(2) loading~~ in response to a second load instruction loads a second ~~source vector~~ plurality of data bytes into a second register, and [[;]]

~~(3) reading~~ in response to a shuffle instruction reads a first plurality of data elements from ~~said~~ the first register and a second plurality of data elements from ~~said~~ the second register, [[;]] and

~~(4) writing said~~ writes the first plurality of data elements and ~~said~~ the second plurality of data elements into a third register in a particular order specified by the shuffle instruction to produce a ~~target~~ vector having a plurality of data elements aligned for ~~SIMD~~ single instruction multiple data (SIMD) processing; and

a SIMD processing unit coupled to the first processing unit that operates on the vector.

50. (currently amended) The ~~method as recited in~~ microprocessor of claim 49, wherein ~~said writing step comprises:~~ in response to the shuffle instruction, the first processing unit

~~writing~~ writes ~~even numbered, lower~~ zero-extended data elements of ~~said~~ the first register to ~~said~~ the third register. ~~; and~~

~~writing sign bits of odd numbered, lower elements of said first register to said third register.~~

51. (currently amended) The ~~method as recited in~~ microprocessor of claim 49, wherein ~~said writing step comprises:~~ in response to the shuffle instruction, the first processing unit

~~writing even numbered, upper elements of said first register to said third register; and~~

~~writing~~ writes ~~sign bits of odd numbered, upper~~ sign-extended data elements of ~~said~~ the first register to ~~said~~ the third register.

52. (currently amended) ~~A method for generating an ordered set of elements in a target vector from elements in a first source vector and a second source vector for single instruction multiple data (SIMD) vector processing~~ The microprocessor of claim 49, ~~comprising the steps of:~~ wherein in response to the shuffle instruction, the first processing unit

~~(1) loading the first source vector into a first register;~~

~~(2) loading the second source vector into a second register;~~

~~(3) selecting~~ reads a first subset of data elements from ~~said~~ the first register, ~~said~~ the first subset of data elements comprising any one of the following groups of data elements from the first ~~source vector~~ register: odd elements, even elements, lower elements and upper elements; and

~~(4) selecting~~ reads a second subset of data elements from ~~said~~ the second register, ~~said~~ the second subset of data elements comprising any one of the following groups of data elements from the second ~~source vector~~ register: odd elements, even elements, lower elements and upper elements.

53. (currently amended) The ~~method~~ microprocessor of claim 52, ~~further comprising the step of:~~ wherein in response to the shuffle instruction, the first processing unit ~~(5) writing said~~ writes the first subset of data elements and ~~said~~ the second subset of data elements into [[a]] the third register. ~~to facilitate a particular SIMD vector processing operation, said first subset being written into any one of the following groups of elements in said third register: upper elements, odd elements, and odd elements in reverse order, and said second subset being written into any one of the following groups of elements in said third register: lower elements, even elements, and even elements in reverse order, wherein elements written into said third register comprise the target vector.~~

54. (canceled)

55. (currently amended) The ~~method as recited in~~ microprocessor of claim 49 ~~54~~, wherein the vector in the third register ~~is comprised of~~ comprises a plurality of ~~eight~~ 8-bit data elements.

56. (currently amended) The ~~method as recited in~~ microprocessor of claim 49 ~~54~~, wherein the vector in the third register ~~is comprised of four~~ comprises a plurality of 16-bit data elements.

57. (currently amended) The ~~method as recited in~~ microprocessor of claim 1, wherein the starting data byte is specified ~~as~~ by a variable in a ~~register in an~~ field of the alignment instruction.

58. (currently amended) The ~~method as recited in~~ microprocessor of claim 1, wherein the first ~~vector~~ plurality of data bytes and the second ~~vector~~ plurality of data bytes are ~~in~~ loaded from contiguous locations ~~in the~~ of a memory unit.

59. (currently amended) The ~~method as recited in~~ microprocessor of claim 1, wherein the ~~processor~~ first processing unit operates in a big-endian byte ordering mode.

60. (currently amended) The ~~method as recited in~~ microprocessor of claim 1, wherein the ~~processor~~ first processing unit operates in a little-endian byte ordering mode.

61. (canceled)

62. (canceled)

63. (canceled)

64. (canceled)

65. (new) A microprocessor having a plurality of registers, comprising:

a first processing unit that

in response to an alignment instruction determines a starting data byte in a first register specified by the alignment instruction, wherein the starting data byte specifies a first data byte of a first vector, extracts the first vector from the first register specified by the alignment instruction and a second register specified by the alignment instruction beginning from a first bit in the starting byte of the first register specified by the alignment instruction continuing through bits in the second register specified by the alignment instruction, and replicates the first vector into a third register specified by the alignment instruction such that the third register specified by the alignment instruction contains a plurality of data elements aligned for single instruction multiple data (SIMD) processing; and

in response to a shuffle instruction reads a first plurality of data elements from a first register specified by the shuffle instruction and a second plurality of data elements from a second register specified by the shuffle instruction, and writes the first plurality of data elements and the second plurality of data elements into a third register specified by the shuffle instruction, in a particular order specified by the shuffle instruction, to produce a second vector aligned for SIMD processing; and

a SIMD processing unit coupled to the first processing unit that operates on vectors aligned for SIMD processing.

66. (new) The microprocessor of claim 65, wherein in response to the shuffle instruction, the first processing unit writes zero-extended data elements of the first register to the third register.

67. (new) The microprocessor of claim 65, wherein in response to the shuffle instruction, the first processing unit writes sign-extended data elements of the first register to the third register.

68. (new) The microprocessor of claim 65, wherein the starting data byte is specified by a variable in a field of the alignment instruction.

69. (new) The microprocessor of claim 65, wherein the first processing unit operates in a big-endian byte ordering mode.

70. (new) The microprocessor of claim 65, wherein the first processing unit operates in a little-endian byte ordering mode.